EP - 4 .   Require func . → include a module to other

Module's protect their
variables & func. from
leaking .

`require ( ./ xyz.js )`

const ───
```
require(app.js)
require(sum.js)

clg("in index");
```
index.js

```
clg("in app")
```
app.js

```
clg("in sum)
func(){

  3.
module.export = func name.
```
sum.js
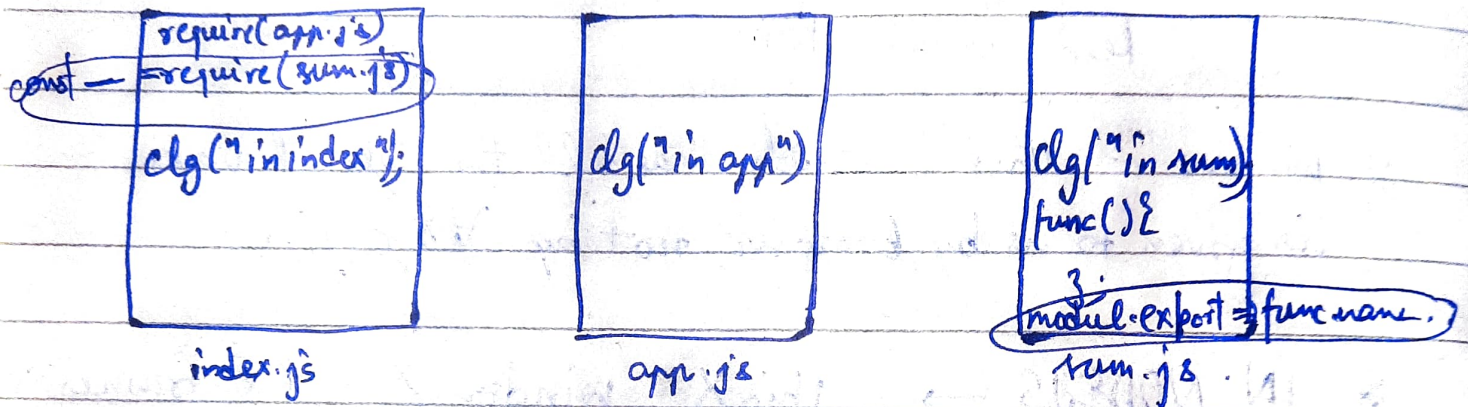
Output :   > in app. js .
in sum.js .
" func() is not defined." ERROR.

#★
By default, func & vars
are protected .

#★
we have to explicitly
import & export module

Why this protection over modules is needed ??

↳   if same var names are created, so there colld
be conflicts b/w files

There are 2 type of Module patterns.

Common JS Modules. (cjs)	ES Modules. (mjs)
→ module. exports require ();	→ export import
→ by default used by nodejs	→ by default in react & etc.
→ Older way	→ Newer way.
→ Synchronous way	→ Asynchrous way.
→ nonstrict mode	→ strict mode ↳ can't declare variable without defining there datatype

NOTE,
→ module. export is an empty Object.

```
module.export.x= x;
module.export. calculateSum = calculateSum;
```